SProt - From Local to Global Protein Structure Similarity

Jakub Galgonek, David Hoksza
Department of Software Engineering
Faculty of Mathematics and Physics, Charles University in Prague
Prague, Czech Republic
Email: {galgonek, hoksza}@ksi.mff.cuni.cz

***Abstract*—Similarity search in protein databases is one of the most essential issues in proteomics. With the growing number of experimentally solved protein structures, the focus shifted from sequence to structure. The area of structure similarity forms a big challenge since even no standard definition of optimal similarity exists in the field. In this paper, we propose a protein structure similarity method called SProt. SProt concentrates on high-quality modeling of local similarity in the process of feature extraction. SProt's features are based on spherical spatial neighborhood where similarity can be well defined. On top of the partial local similarities, global measure assessing similarity to a pair of protein structures is built. SProt outperforms other methods in classification accuracy, while it is at least comparable to the best existing solutions in terms of precision-recall or quality of alignment.**

***Keywords*-Protein structure similarity; TM-score; RMSD;**

I. Introduction

Proteins play a key role in all living organisms. If we consider DNA as the program of life, then protein network can be understood as a living computer that protects and interprets this program. The connection between the program and the computer goes much further since big parts of this program (so-called codding sequences of DNA) include construction plans for every component of the computer (i.e. prescripts how to synthesize proteins).

Both DNA and proteins can be represented by chemical sequences consisted of given types of building blocks. DNA is constructed from 4 types of nucleotides whereas proteins consist of 20 types of amino acids.

The function of a protein is consequence of its spatial conformation rather than of ordering of its amino acids (protein sequence). Thus, the protein structure is closer to the function than the sequence being the reason for enormous effort spent on protein structure research. Moreover, the biological motivation for protein structure similarity search stems from the thesis that proteins having similar structures also share similar function. Hence, it is very useful to have tools for measuring protein structure similarity in order to be able to identify similar protein structures from a database of protein structures with already known function.

II. Motivations

Most of the protein structure similarity measures are based on comparisons of positions of amino acids in the space. For this purpose, amino acids are represented as coordinates of their α-carbon (and sometimes β-carbon) atoms. The protein structure similarity assessment usually comprises two steps. In the first one, which we call *alignment search*, amino acid inter-protein pairing is established. The second step, which we call *superposition search*, comprehends superposition minimizing the selected similarity/distance function. This function usually aggregates values based on distances of the paired amino acids after the superposition.

Although it has been shown that the structure similarity problem, consisting of alignment and superposition search, is NP-hard [1], each of the parts of the problem can be solved in polynomial time using the knowledge of the other part. Thus, when we have the knowledge of the alignment, there exist methods how to obtain superposition optimizing given similarity/distance formula in polynomial time (e.g. *Kabsch* algorithm [2] for RMSD). On the other hand, if we are provided with the superposition and the measure is in form of a sum function, we can use dynamic programming to determine the respective optimal alignment with respect to the given superposition. The dynamic programming has to employ scoring corresponding to the inner part of the aggregation (sum) function. E.g. for RMSD, the score for i-th and j-th amino acids of the superposed proteins equals their squared distance in Euclidean space. Thus, the structure similarity is a recursive problem where both parts needed for optimal solution are mutually dependent and at the beginning we know neither the alignment, nor the superposition.

III. Existing solutions

In this section, we briefly describe the main ideas of the algorithms which we see as the state-of-the-art solutions or solutions which outperform the standard ones and which our presented algorithm will be compared to — *BLAST*, *CE*, *DALI*, *db-iTM*, *PPM*, *Vorolign*, *Vorometric*, *ProtDex2*, *PSI-BLAST*, *3D-BLAST*.

One of the first solutions to protein structure similarity assessment was *DALI* [3], [4] which represents a three-dimensional structure by two-dimensional matrix of inter-residual distances. Similar structures should also share similar distance distribution and thus in comparison process, the matrices are split into overlapping parts and similar (contact) patterns are stored. These are further extended

to obtain the alignment. Similarly to DALI, *Protdex* [5] uses intra-residual distance matrices. But instead of chaining the contact patterns, Protdex splits them to constant-sized submatrices which, together with their description, are used as index terms for inverted file index. Query structures are processed in the same way and the index, together with subsequent scoring, is utilized to identify similar database structures. On the returned list, subsequent refinement is possible with arbitrary alignment-based algorithm. *CE* [6] uses the concept of aligned fragment pairs (AFP) for sufficiently structurally similar portions of the sequence. A few seeds are chosen and iteratively extended by other AFPs where three different measures are taken into account when deciding whether a new AFP should be added to the chain. At the end, a final optimization takes place which chooses the best alignment. *SSAP* [7] heavily exploits Smith-Waterman dynamic programming algorithm [8]. Each residue is represented by distances to every other residue. For each pair of amino acids in the compared structures dynamic programming matrix is computed with scoring matrix based on these distances (local similarity). In the second level dynamic programming, these matrices are aggregated to obtain the resulting structure alignment (global similarity). More recently, methods based on Voronoi diagrams were proposed. *Vorometric* [9] forms contact strings from the Delaunay tessellation and these are stored in a metric index. For finding similar contact string with the query, edit distance with metric scoring matrix is used. Found hits are used as seeds for consequent step, where modification of dynamic programming is applied to the hits to obtain the alignment. *Vorolign* [10] extracts nearest-neighbor sets for each amino acid based on the Voronoi tessellation. Similarity of the sets is defined, which is further used in dynamic programming assessing similarity to a pair of amino acids. Local similarities are used as scores for second-level dynamic programming. The same collective of authors later introduced a solution called *PPM* [11]. PPM identifies sufficiently similar (core) blocks which are then used to create a graph of core blocks. That path in the graph is chosen, that minimizes the cost of mutation. *db-iTM* [12] is a recently proposed solution which represents amino acids as set of concentric circles and based on their densities and radii forms feature vectors which are used in local dynamic programming. Last of the structure-based methods presented in this overview is *3D-BLAST* [13] which derives structural alphabet from the $\kappa - \alpha$ plot and structures represented as strings over this alphabet are accessed using the BLAST approach. That takes us to the last two methods which are purely sequence-based and thus are able to provide comparison of structure-based approaches with the sequence-based ones - *BLAST* [14] and *PSI-BLAST* [15]. BLAST is a state-of-the-art tool for similarity search in protein sequence databases and is based on heuristic which noticeably decreases runtime needed for full Smith-Waterman algorithm [8] being the optimal measure for assessing similarity to a pair of protein structures. PSI-BLAST extends the original BLAST algorithm by employment of the position-specific score matrix and is more sensitive to weak sequential similarities.

IV. SProt

In contrast to most of the presented algorithms, in our solution we put a lot of emphasis on high-quality modeling of local similarities of the amino acids. We believe that representing proteins by various derived features might cause loss of information which is inevitable for quality alignment. In this section we present our solution, called *SProt*, which aims to avoid the possible loss-of-information drawback.

A. Basic Idea

Determining alignment and superposition of protein structures is, as we already mentioned, a nontrivial problem. However, what holds true for whole protein structures, does not have to be valid for small substructures. If we want to align two small parts of two protein backbones, the natural way is to execute gapless alignment for these parts. When aligning only few amino acids it does not make sense to introduce gaps and thus the alignment is defined unambiguously. We further employ this alignment in consequent step, where we add to the alignment amino acids being spatially close to the already aligned backbone amino acids (which do not have to be those being close in sequence order). In this way, we are able to take spatial neighborhood into account when modeling local similarity. As stated above, given an alignment, to compute the superposition is a relatively easy task. The outlined principle is the central point of high-quality local modeling of the SProt method.

In the following sections we describe details of the algorithm but before that we briefly present its overview. SProt represents each amino acid A as a small substructure containing all amino acids that are spatially close to A (section IV-B). Alignment and superposition are subsequently used to express similarity between such representations of amino acid (section IV-C) use of which we justified above. The computed local similarity between pairs of amino acids is used in dynamic programing method to obtain global structural alignment. Quality of this alignment, that represents the resulting structure similarity, is expressed in terms of value of TM-score (section IV-D).

B. Representation of Proteins

Each amino acid A is represented by the amino acid content of the Euclidean sphere with center in the A and given radius. Since the characterization of A is based on its close spatial-based amino acid neighborhood bounded by the Euclidean sphere, we call the representation *aa-sphere*.

SProt represents each amino acid by its α-carbon. However, when testing intersection of an amino acid with a sphere, all heavy atoms of the amino acid are considered, not

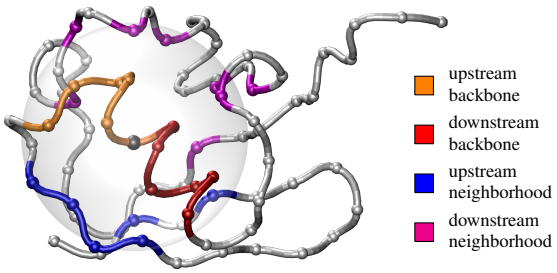

Figure 1. The aa-sphere of the 26-th amino acid of the $1ubq$ protein. Each amino acid is represented by a ball centered in its α-carbon position. The tube corresponds to the protein backbone denoting the protein sequence. The Euclidean sphere with center in the 26-th amino acid (denoted in black color) and with radius 9 Å is shown in gray. The different colors emphasize amino acids included in the aa-sphere. The colored amino acids having its α-atom outside the sphere intersect some of its heavy atoms with the sphere and thus are included in the aa-sphere.

only the α-carbon. Such an approach allows us to include amino acids into the aa-sphere whose α-carbons are too far from the aa-sphere's center but their side chains are still close enough. These amino acids are thus still relevant.

We divide the content of each aa-sphere into three categories:

- *spherical backbone* is the maximal continual part of the amino acid sequence that is included in the aa-sphere and contains the central amino acid. A spherical backbone is divided into *upstream spherical backbone* for amino acids that precede the central amino acid in protein sequence and *downstream spherical backbone* for amino acids that follow the central amino acid in sequence.
- *upstream neighborhood amino acids* contains amino acids in the aa-sphere that preclude the central amino acid in protein sequence and are not included in the spherical backbone.
- *downstream neighborhood amino acids* contains amino acid in the aa-sphere that follow the central amino acid in protein sequence and are not included in the spherical backbone.

Example of an aa-sphere with denoted categories depicts Figure 1 (generated by VMD [16]).

For the purposes of the following steps (namely dynamic programming), amino acids in each category keep the original protein sequence ordering. We also define the term *quantity characteristic* for each aa-sphere to denote size of a particular, above introduced, category. The summary of these characteristics is called *quantity characteristics* of an aa-sphere. Whole protein can than be modeled as a sequence of aa-spheres built for every amino acid.

C. Sphere Similarity

We measure similarity of a pair of aa-spheres using alignment and superposition which is altogether a simple problem for spheres, as we already mentioned. The similarity assessment for aa-spheres consists from three steps:

1) *generating seed spherical backbone alignment.* Spherical backbones are aligned using gapless alignment. The alignment is unique since it is gapless and the central amino acids are aligned to each other.
2) *computing spherical backbone superposition.* The seed alignment dictates spherical superposition being carried out by Kabsch algorithm [2] with linear complexity.
3) *generating spherical alignment.* In the previous step, we have superposed the spherical backbones, but to asses similarity to whole aa-spheres we also have to consider the rest of them. Therefore, the obtained superposition is used to align rest of the amino acids in the aa-sphere (upstream/downstream neighborhood). We apply Needleman-Wunsch algorithm [17] (global alignment) on remaining amino acids in upstream and downstream separately. The algorithm uses scoring function in the form:
$$S_{ij} = \frac{1}{1 + \left(\frac{d_{ij}}{d_s}\right)^2} \tag{1}$$
where d_{ij} is distance of i-th and j-th amino acids according to the spherical superposition and d_s represents a scale parameter (empirically determined).
4) *computing raw spherical measure (SM-raw).* Raw spherical measure/score for aa-spheres x and y is computed for the whole spherical alignment A (steps 1, 2, 3) as:
$$\text{SM-raw} = \sum_{i=1}^{L_A} \frac{1}{1 + \left(\frac{d_i}{d_s}\right)^2} \cdot \frac{1}{max_{[x][y]}} \tag{2}$$
where L_A is length of A, d_i is distance between i-th pair of amino acids according to the spherical superposition, d_s is the same scale parameter used in the previous step and $max_{[x][y]}$ is the maximal possible value of the sum in the numerator for the aa-spheres with quantity characteristics equal to quantity characteristics of aa-spheres x and y (normalization).
5) *computing normalized spherical measure.* The value of SM-raw that we can expect to occur only by chance highly depends on quantity characteristics of compared aa-spheres due to the fact that better superpositions are more probable for smaller aa-spheres. This presents a problem for comparisons of similarity between pairs of spheres with different quantity characteristics. Hence, we computed the empirical cumulative distribution function (ECDF) for SM-raw, specific to given quantity characteristics of compared aa-spheres x and y (denoted as $\widehat{F}_{[x][y]}$) to transform SM-raw to normalized score. Using ECDF allows us to express the probability that a better result could not be obtained by chance for aa-spheres with identical quantity characteristics.

However, such a modification is not sufficient. If, for example, aa-sphere B is obtained from aa-sphere A by removing some amino acids, then SM-raw for these two aa-spheres will be maximum for given quantity characteristics. It implies that ECDF of SM-raw will be maximal too, which is not correct. Hence, factor capturing differences in quantity characteristics of compared aa-spheres x and y was added:

$$f(x,y) = \frac{\sum\limits_{q \in \{q_{ub}, q_{db}, q_{un}, q_{dn}\}} min(q(x), q(y)) + 1}{\sum\limits_{(s,t) \in \{(q_{ub}, q_{un}), (q_{db}, q_{dn})\}} max(s(x) + t(x), s(y) + t(y)) + 1} \tag{3}$$

where q_{ub}, q_{db}, q_{un} and q_{dn} denote individual quantity characteristics of an aa-sphere.

Full normalized score for aa-spheres x and y has then following form:

$$\text{SM-score}(x,y) = f(x,y)\widehat{F}_{[x][y]}(\text{SM-raw}(x,y)) \tag{4}$$

D. Alignment and Superposition

The spherical similarity measure is employed in generation of global alignment of the compared protein structures using logarithm of SM-score as a scoring function and linear gap penalty model. SM-score estimates probability that matching given pairs of spheres is significant, thus logarithm of this value used inside the Needleman-Wunsch algorithm maximizes probability that the resulting alignment is significant.

After obtaining alignment, we utilize well-established TM-score superposition algorithm [18] to get the superposition and score. TM-score algorithm is designed to maximize following similarity formula:

$$\text{TM-score} = \frac{1}{L_T} \sum_{i=1}^{L_A} \frac{1}{1 + \left(\frac{d_i}{d_0(L_T)}\right)^2} \tag{5}$$

where L_A is length of the alignment A, L_T length of the query protein, d_i distance between i-th pair of amino acids according to the superposition computed by the TM score algorithm and $d_0(L_T)$ is a scale function.

V. Optimizations

The proposed algorithm depends on several parameters, that must be suitably set to obtain high-quality results.

1) sphere radius: The parameter determines number of amino acids in an aa-sphere. Small radius leads to low number of amino acids in an aa-sphere which in turns causes loss of accuracy. On the other hand, using large radius increases the time needed for computing sphere similarity since it influences runtime of the Needleman-Wunsch algorithm which is of quadratic complexity.

In our experimental section we used sphere radius 9 Å as a trade-off between time and accuracy.

2) scale parameter d_s: Spherical similarity measure is a variant of TM-score which uses scale parameter dependent on the size of the compared proteins. However, TM-score's parametrization is not suitable for aa-spheres, because they are smaller than protein sequences. Therefore, we used constant value scale parameter as predecessors of TM-score did. For example, MaxSub [19] used value 3.5 Å, S-score [20] used value 5 Å. We decided to set the parameter to 2 Å due to the generally smaller sizes of aa-spheres in comparison to average protein size.

3) raw-score empirical cumulative distribution function: The empirical cumulative distribution function (ECDF) of SM-raw score was produces from comprehensive comparison of proteins from ASTRAL-25 v1.65 database [21]. Since the ECDF computation is highly space consuming if every possible combination of quantity characteristics has to be taken into account, sampling was introduced to decrease the space complexity. Upstream and downstream neighborhoods were downsampled by factor of 2, backbone sizes 0 and 1 were treated identically as well as each quantity characteristics exceeding value 7.

4) gap penalty: Setting gap penalty value has essential influence on quality of the measure. We used $log(0.75)$ as the gap penalty value which has the best results for most of the evaluations. This setting of gap penalty is low enough, thus only amino acids with significant similarity will be paired.

VI. Experimental Evaluation

In order to evaluate quality of the proposed measure, we focus on expressing how well the measure fits human intuitive view of protein structure similarity. However, difficulty of this task lies in the absence of a large-scale human-curated database of pairwise protein structure similarities, which we could use as a standard of truth. However, there exists manually curated hierarchical evolutionary classification SCOP [22] utilizable for this purpose. Using SCOP, we are able to (indirectly) compare SProt with domain expert's conception of the structure similarity. SCOP hierarchy consists of four levels - *family*, *superfamily*, *fold* and *class*. Therefore, SCOP can provide us with the information whether two protein structures are considered similar or not (on the given level) by human observer. Although such a binary measure (similar or dissimilar) is not able to express qualities of the similarity measure such as quality of alignment or superposition, but it is suitable to express performance of the measure in terms of ability of classification and information retrieval.

A. Protein classification

Automatic classification of protein structures is one of the traditional problems in this area. The task is to determine SCOP classification of a query protein according to the investigated measure. The category of the query protein is

Table I
CLASSIFICATION OF VOROLIGN DATASET

Method	Family	Superf.	Fold	RMSD	Cover	TM
SProt	90.4	**96.9**	**98.6**	4.14	81.1	0.63
Vorometric	**90.7**	94.9	97.6	2.43	**87.2**	0.74
PPM	88.3	94.5	97.5	?	?	?
db-iTM	86.6	95.8	98.2	?	?	?
Vorolign	86.4	92.4	97.7	**1.90**	76.3	0.74
CE	84.6	91.9	94.1	1.95	78.2	**0.77**
BLAST	48.9	52.5	52.8	-	-	-

derived from category of the database protein being most similar to the query. Accuracy of classification is measured on given level as the percentage of correctly classified queries.

We used the dataset that was first introduced for evaluation of the *Vorolign* method (*Vorolign dataset*). The database set utilizes ASTRAL-25 v1.65 containing 4,357 structures. 979 structures from difference set between SCOP v1.67 and v1.65 are used as the query set.

Results on this datasets summarizes Table I. The table describes the classification accuracy for family, superfamily and fold levels. It also shows average values of several characteristics describing the algorithms from different points of view. Namely, the table contains average TM-score, average RMSD and average alignment cover (i.e., how many percent of amino acids of a query is aligned) between each query and its most similar structure used for classification. On the superfamily and fold level, SProt outperforms the other solutions, while on the family level SProt is slightly outmatched by Vorometric. It is interesting to realize that although the other solutions stand out in terms of average values of the various characteristics, SProt outperforms them in terms of classification accuracy. Thus, better partial characteristics do not necessary lead to better real-world results.

B. Information retrieval in protein structure databases

In the previous section, we measured the hit rate based on the most similar database structure. Thus, the most similar structure was the only determinant of the quality. However, often the user wants to obtain all relevant structures not only the most similar one. Result can then be visualized as a list of database structures ordered according to the given measure with the most similar structure on top. Correctness of such ordering can be measured in terms of precision and recall. Precision expresses how many percent of structures at the given cut-off rank of the result list are relevant. Recall expresses how many of all relevant results are obtained at the given cut-off rank of the result list. The precision-recall dependence can be expressed in a graph that describes the average precision of queries for different recall levels. As single-value metric, it is possible to define average precision.

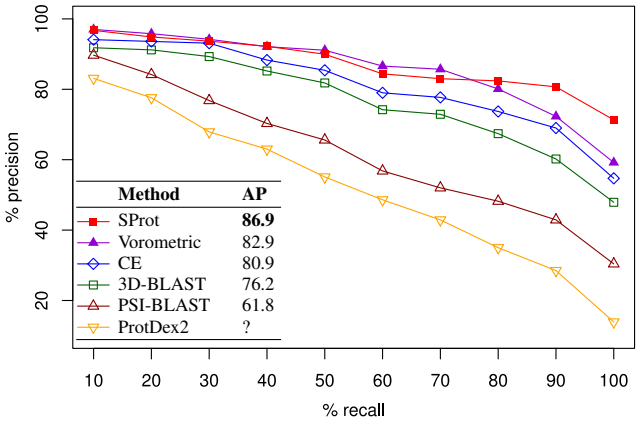

Figure 2. Average precision-recall curves and average precision (AP) for the Protdex dataset.

For this experiment we used the *Protdex dataset* consisting of 34,055 proteins that have been first used for evaluation of the ProtDex2 method. 108 structures from medium-size families of the database set were selected as queries.

We consider a fetched database structure as relevant, if it comes from the same SCOP family as the query. Precision-recall graph and average precision for the used dataset shows Figure 2. The proposed measure has better precision-recall curve than other methods except Vorometric. In comparison with Vorometric, the curve of the proposed measure is slightly worse for medium recall levels while it is noticeably better for high levels. When measuring average precision, SProt outperforms the other methods.

C. Quality of the structural alignments

We have already mentioned that it would also be appropriate to investigate what is the quality of alignments and scores the measure produces. For this purpose the 10 difficult pairs of structures were introduced in [23]. It is obvious from Table I, that SProt does not produce high alignment cover and TM-score. However, to produced better alignment and TM-score it is possible to apply iterative improvement of TM-score. In this case, the superposition obtained by the proposed original measure is used to produce a new better alignment. Such an approach utilize also other methods, e.g. Vorometric. For the purpose of the improvement, Needleman-Wunsch algorithm is used with scoring function:

$$S_{ij} = \begin{cases} \frac{1}{1+\left(\frac{d_{ij}}{d_0(L_T)}\right)^2} & \text{if } d_{ij} < 3d_0(L_T) \\ -\infty & \text{otherwise} \end{cases} \quad (6)$$

where d_{ij} represents distance between i-th and j-th amino acid according to the superposition, L_T length of the query protein and $d_0(L_T)$ the scale function used in TM-score. The $3d_0(L_T)$ threshold is used to prevent aligning too distant amino acids. The resulting alignment is then used in the TM-score algorithm to obtain new score and superposition. This procedure repeats until better score is obtained.

Table II
COMPARISON OF ALIGNMENT QUALITY ON 10 PAIRS

Method	RMSD	Cover	TM
SProt + TM-optimization	3.29	85.8	**0.65**
SProt	7.29	73.8	0.43
Vorometric	3.02	84.8	**0.65**
Vorolign	**2.28**	51.7	0.56
DaliLite	2.82	80.0	0.61
SSAP	4.37	**88.1**	0.59
CE	3.17	83.4	0.60

As shown in Table II, this approach significantly improves the cover and score. On the other hand, extensive use of the iterative concept does not improve the results of the previous evaluations whereas it noticeably downgrades performance of the algorithm. The original SProt algorithm (without employing the iterative improvement procedure) takes 6.4 min on Vorolign dataset per query and 77.8 min on Protdex dataset per query recalculated for one core on a machine with two processors Intel Xeon Quad-Core E5345 2.33 GHz and 8 GB RAM. Thus we can see that the time is an issue in our case.

VII. CONCLUSION

We proposed a novel algorithm that puts emphasis on high-quality modeling of local similarities of the amino acids. That is achieved by representing each amino acid by its spatial-based amino acid neighborhood. This approach leads to good real-world results, especially for superfamily/fold classification accuracy and for precision on high recall levels where we outperform all existing solutions.

ACKNOWLEDGMENT

The work was supported by Czech Science Foundation project Nr. 201/09/0683 and by the grant SVV-2010-261312.

REFERENCES

[1] R. H. Lathrop, "The protein threading problem with sequence amino acid interaction preferences is NP-complete," *Protein Eng*, vol. 7, no. 9, 1994.

[2] W. Kabsch, "A solution for the best rotation to relate two sets of vectors," *Acta Crystallogr A*, vol. 32, no. 5, 1976.

[3] L. Holm, "Protein structure comparison by alignment of distance matrices," *J Mol Biol*, vol. 233, no. 1, 1993.

[4] L. Holm and J. Park, "DaliLite workbench for protein structure comparison." *Bioinformatics*, vol. 16, no. 6, 2000.

[5] Z. Aung and K. L. Tan, "Rapid 3D protein structure database searching using information retrieval techniques." *Bioinformatics*, vol. 20, no. 7, 2004.

[6] I. N. Shindyalov and P. E. Bourne, "Protein structure alignment by incremental combinatorial extension (CE) of the optimal path." *Protein Eng*, vol. 11, no. 9, 1998.

[7] W. Taylor, T. Flores, and C. Orengo, "Multiple protein structure alignment," *Prot Sci*, vol. 3, no. 10, 1994.

[8] T. F. Smith and M. S. Waterman, "Identification of common molecular subsequences." *J Mol Biol*, vol. 147, no. 1, 1981.

[9] A. Sacan, H. I. Toroslu, and H. Ferhatosmanoglu, "Integrated search and alignment of protein structures," *Bioinformatics*, vol. 24, no. 24, 2008.

[10] F. Birzele, J. E. Gewehr, G. Csaba, and R. Zimmer, "Vorolign–fast structural alignment using Voronoi contacts." *Bioinformatics*, vol. 23, no. 2, 2007.

[11] G. Csaba, F. Birzele, and R. Zimmer, "Protein structure alignment considering phenotypic plasticity," *Bioinformatics*, vol. 24, no. 16, 2008.

[12] D. Hoksza and J. Galgonek, "Density-based classification of protein structures using iterative TM-score," in *BIBMW: 2009 IEEE International Conference on Bioinformatics and Biomedicine Workshop*. IEEE, 2009, Proceedings Paper.

[13] C.-H. H. Tung, J.-W. W. Huang, and J.-M. M. Yang, "Kappa-alpha plot derived structural alphabet and BLOSUM-like substitution matrix for rapid search of protein structure database." *Genome biology*, vol. 8, no. 3, 2007.

[14] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool." *J Mol Biol*, vol. 215, no. 3, 1990.

[15] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman, "Gapped BLAST and PSI-BLAST: a new generation of protein database search programs." *Nucl. Acids Res.*, vol. 25, no. 17, 1997.

[16] W. Humphrey, A. Dalke, and K. Schulten, "VMD: visual molecular dynamics." *J Mol Graph*, vol. 14, no. 1, 1996.

[17] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins." *J Mol Biol*, vol. 48, no. 3, 1970.

[18] Y. Zhang and J. Skolnick, "Scoring function for automated assessment of protein structure template quality," *Proteins*, vol. 57, no. 4, 2004.

[19] N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer, "MaxSub: an automated measure for the assessment of protein structure prediction quality." *Bioinformatics*, vol. 16, no. 9, 2000.

[20] S. Cristobal, A. Zemla, D. Fischer, L. Rychlewski, and A. Elofsson, "A study of quality measures for protein threading models," *BMC Bioinformatics*, vol. 2, no. 1, 2001.

[21] J.-M. M. Chandonia, G. Hon, N. S. Walker, L. Lo Conte, P. Koehl, M. Levitt, and S. E. Brenner, "The ASTRAL compendium in 2004." *Nucleic Acids Res.*, vol. 32, 2004.

[22] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: a structural classification of proteins database for the investigation of sequences and structures." *J Mol Biol*, vol. 247, no. 4, 1995.

[23] D. Fischer, A. Elofsson, D. Rice, and D. Eisenberg, "Assessing the performance of fold recognition methods by means of a comprehensive benchmark." in *Pac. Symp. Biocomput*, 1996.